05-30-00

A

MILDE, HOFFBERG & MACKLIN, LLP

COUNSELORS IN INTELLECTUAL PROPERTY LAW
10 BANK STREET, SUITE 460
WHITE PLAINS, NY 10606

TEL. (914) 949-3100
FAX (914) 949-3416
INTPROP@AOL.COM

[illegible]. MILDE, JR.
STEVEN M. HOFFBERG*
OF COUNSEL:
KENNETH E. MACKLIN†
PAUL S. HOFFMAN
SAMUEL SHIPKOVITZ, Ph.D.‡

*ALSO ADMITTED IN CT
†ALSO ADMITTED IN DC
‡ADMITTED IN DC, VA, OH ONLY

FORM PTO-1082 (modified)

Case Docket No.: BUFFALO 201

THE COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, DC 20231

Sir:

Date: May 26, 2000

Transmitted herewith for filing is the patent application of

Inventor(s): David Friedman
Wai Wu

For: VOICE OVER INTERNET CALL CENTER INTEGRATION

Enclosed are:

[X] Specification - 27 pages

[x] Claims - 4 pages

[x] Abstract - 1 page

[ ] Sheets of drawing. (FIGS.)

[ ] A certified copy of a ________ application (priority document).

[x] A declaration and power of attorney.

[x] Assignment

[x] Assignment recordation cover sheet

[x] Check for $40.00 Assignment recordation fee

[x] Appendix - 27 pages

[x] Verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27

[ ] Priority is hereby claimed on the basis of the following:

Country	Serial No.	Date

The filing fee has been calculated as shown below:

FOR:	(Col. 1) NO. FILED	(Col. 2) NO. EXTRA	SMALL ENTITY RATE	 FEE	OTHER THAN A SMALL ENTITY OR	 RATE	 FEE	
BASIC FEE				$ 345.00	OR		$ 690.00	
TOTAL CLAIMS	21 - 20=	1	x 9=	$ 9.00	OR	x 18=	$	
INDEP CLAIMS	4 - 3=	1	x 39=	$ 39.00		OR	x 78=	$
[ ] MULTIPLE DEPENDENT CLAIM PRESENTED			+130=	$	OR	+260=	$	
*If the difference in Col. 1 is less than zero, enter"0" in Col. 2			TOTAL	$ 393.00	OR		$	

[x] A check in the amount of $ 393.00 to cover the filing fee is enclosed.

[x] Please charge any insufficiency of fee, or credit any excess, to Deposit Account No. 50-0427.

MILDE, HOFFBERG & MACKLIN, LLP

By____________________
Steven M. Hoffberg
Reg. No. 33,511

914-949-3100

"Express Mail" mailing label.

Number EK677256746US
Date of Deposit May 26, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, DC 20231.

By: ____________________

Applicants or Patentees: Friedman, et al. Attorney's
Serial or Patent No.: ______ Docket No.: BUFFALO 2
Filed or Issued: ______
For: VOICE OVER INTERNET CALL CENTER INTEGRATION

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9(f) and 1.27 (b)) - INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees under section 41(a) and (b) of Title 35, United States Code, to the Patent and Tradema Office with regard to the invention entitled VOICE OVER INTERNET CALL CENTER INTEGRATION described

[x] the specification filed herewith
[ ] application serial no. ______ filed ______
[ ] patent no. ______ issued ______

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor und 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obliga under contract or law to assign, grant, convey, or license any rights in the invention is listed below:

[ ] no such person, concern, or organization
[x] persons, concerns organizations listed below*

*NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

FULL NAME Buffalo International, Inc.
ADDRESS 400 Columbus Avenue, Valhalla, NY 10595
[ ] INDIVIDUAL [x] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATIO

FULL NAME ______
ADDRESS ______
[ ] INDIVIDUAL [ ] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATIO

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenan fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Ti 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

David Friedman	Wai Wu
NAME OF INVENTOR	NAME OF INVENTOR
Signature of Inventor	Signature of Inventor
5/26/00	5/26/00
Date	Date

Applicants or Patentees: Friedman, et al. Attorney's
Serial or Patent No.: Docket No.: BUFFALO 201
Filed or Issued:
Title: VOICE OVER INTERNET CALL CENTER INTEGRATION

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS (37 CFR 1.9 (f) and 1.27(c) - SMALL BUSINESS CONCERN

I hereby declare that I am

[ ] the owner of the small business concern identified below:
[x] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Buffalo International, Inc.
ADDRESS OF SMALL BUSINESS CONCERN 400 Columbus Avenue
Valhalla, NY 10595

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent Office in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[x] the specification filed herewith with title as listed above.
[ ] the application identified above.
[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights to the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person concern or organization having any rights in the invention is listed below:
[ ] no such person concern or organization exists.
[x] each such person concern or organization is listed below.

Separate verified statements are required from each named person concern or organization having rights to the invention averring to their status as small entities (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING David Friedman
TITLE OF PERSON OTHER THAN OWNER Vice President
ADDRESS OF PERSON SIGNING 400 Columbus Avenue
Valhalla, NY 10595

SIGNATURE DATE 5/26/00

VOICE OVER INTERNET CALL CENTER INTEGRATION

FIELD OF THE INVENTION

The present invention relates to the field of voice over Internet Protocol (VOIP) communications systems, and more particular to new and useful functions implemented using the VOIP associated ability to reach large audiences.

BACKGROUND OF THE INVENTION

Much information on the nature of traditional computer telephony hardware and software is available in a number of publications accessible to the present inventors and to those with skill in the art in general. One document which provides considerable information on intelligent networks is "ITU-T·Recommendation Q.1219, Intelligent Network User's Guide for Capability Set 1", dated April, 1994. This document is incorporated herein by reference.

In a call center, a relatively large number of agents typically handle telephone communication with callers. Each agent is typically assigned to a telephone connected to a central switch, which is in turn connected to a public-switched telephone network (PSTN), well-known in the art. The central switch may be one of several types, such as Automatic Call Distributor (ACD), Private Branch Exchange (PBX), or PSTN. Each agent also typically has access to a computer platform having a video display unit (PC/VDU) which may be adapted, with suitable connectivity hardware, to process Internet protocol telephony calls.

Intelligent telephony networks and IP networks typically share infrastructure to some extent, and computer equipment added to telephony systems for computer-telephony integration (CTI) are also capable of Internet connection and interaction. There is therefore often no clear distinction as to what part of a network is conventional telephony, and what part is IPT. In conventional telephony systems, such as publicly-switched telephony networks (PSTNs), there are computerized service control points (SCPs) that provide central routing intelligence (hence intelligent network). IPNs do not have a central router intelligence, such as a SCP. IPNs, however, have multiple Domain Name Servers (DNS), whose purpose is basically the same as the routers in intelligent networks, which is controlling the routing of traffic. Instead of telephony switches (PBXs), IP switches or IP routers are used. An organization having one or more call centers for serving customers typically provides one or more telephone numbers to the public or to their customer base, or both, that may be used to reach the service. In the case of an IP network, a similar organization may provide an IP address for client access to services, and there are a number of ways the IP address may be provided. Such numbers or addresses may be published on product packaging, in advertisements, in user manuals, in computerized help files, and the like.

Routing of calls in intelligent networks, then, may be on several levels. Pre-routing may be done at SCPs and further routing may be accomplished at individual call centers. As described above a call center in an intelligent telephony system typically involves a central switch The central switch is typically connected to a publicly-switched telephone network (PSTN), well-known in the art. Agents, trained (hopefully) to handle customer service, man telephones connected to the central switch. This arrangement is known in the art as Customer Premises Equipment (CPE).

If the call center consists of just a central switch and connected telephone stations, the routing that can be done is very limited. Switches, although increasingly computerized, are limited in the range of computer processes that may be performed. For this reason additional computer capability in the art has been added for such central switches by connecting computer processors adapted to run control routines and to access databases. The processes of incorporating computer enhancement to telephone switches is known in the art as Computer Telephony Integration (CTI), and the hardware used is referred to as CTI equipment.

In a CTI system telephone stations connected to the central switch may be equipped also with computer terminals, as described above, so agents manning such stations may have access to stored data as well as being linked to incoming callers by a telephone connection. Such stations may be interconnected in a network by any one of several known network protocols, with one or more servers also connected to the network one or more of which may also be connected to a processor providing CTI enhancement, also connected to the central switch of the call center. It is this processor that provides the CTI enhancement for the call center. Agents having access to a PC/VDU connected on a LAN to a CTI processor in turn connected to a telephony switch, may also have multi-media capability, including Internet connectivity, if the CTI processor or another server connected to the LAN provides control for Internet connectivity for stations on the LAN.

When a telephone call arrives at a call center, whether or not the call has been pre-processed at a SCP, typically at least the telephone number of the calling line is made available to the receiving switch at the call center by a telephone carrier. This service is available by most PSTNs as caller-

ID information in one of several formats. If the call center is computer-enhanced (CTI) the phone number of the calling party may be used to access additional information from a database at a server on the network that connects the agent workstations. In this manner information pertinent to a call may be provided to an agent.

Referring now to the example proposed of a technical-service organization, a system of the sort described herein will handle a large volume of calls from people seeking technical information on installation of certain computer-oriented equipment, and the calls are handled by a finite number of trained agents, which may be distributed over a decentralized matrix of call centers, or at a single call center. In examples used herein illustrating various aspects of the present invention, the case of a decentralized system of multiple call centers will most often be used, although, in various embodiments the invention will also be applicable to individual call centers.

Even with present levels of CTI there are still problems in operating such call centers, or a system of such call centers. There are waiting queues with which to contend, for example, and long waits may be experienced by some callers, while other agents may be available who could handle callers stuck in queues. Other difficulties accrue, for example, when there are hardware or software degradations or failures or overloads in one or more parts of a system. Still other problems accrue due to known latency in conventional equipment. There are many other problems, and it is well recognized in the art, and by the general public who have accessed such call centers, that there is much room for improvement in the entire concept and operation of such call center systems. It is to these problems, pertaining to efficient, effective, timely, and cost-effective service to customers (users) of call center systems that aspects and embodiments of the present invention detailed below are directed.

See, United States Patent 6,064,667 (Gisby, et al., May 16, 2000), expressly incorporated herein by reference.

Business applications for call centers are virtually unlimited in the types of transactions that they can accommodate. Call centers can support, for example, sales, including order entry, order inquiry, and reservations; financial services, including funds transfer, credit card verification, and stock transactions; information services, including event schedules, referral services, transportation schedules, and yellow pages; customer services, including technical support, repair dispatch, and claims handling.

Despite the almost innumerable applications, however, existing ACD centers have several limitations. One notable drawback to ACDs is that they lack intelligence. In other words, limited inflexible intelligence is built into the ACD software. Routing of inbound and outbound calls is based on circuit switching. Furthermore, existing ACD centers cannot be accessed through a plurality of access means, such as voice, data, and video. To maximize existing resources and to take advantage of the latest technology, ACD manufacturers are trying to open their systems to third-party inbound and outbound call management systems by integrating, via the recently introduced Computer Telephony Integration ("CTI") standards, the Telephony Application Programming Interface ("TAPI"), the Telephony Services Application Programming Interface ("TSAPI"), and other proprietary protocols. These third-party inbound and outbound call management systems apply computer control and functionality to telephones. Adding computer intelligence to unintelligent telephone devices provides users with more information about inbound calls and lets them use telephones more effectively to distribute information by providing not only customer records coincident with inbound and outbound phone calls, but also skill-based call routing matching agent skills with caller needs and virtual, or geographically distributed, call centers.

One trend in the ACD industry is to enable call agents to be more efficient, productive and to ultimately provide the best service to the customer during the first contact. Technologies such as Automatic Number Identification ("ANI") where the caller phone number is passed to the agent, give agents the opportunity to access information about the caller from corporate databases. The latest CTI technologies play a major role in helping companies respond to industry trends and experience increased productivity and customer service goals by integrating traditional ACDs with computers.

In parallel to the development of ACDs, there are Internet Customer Service Centers ("ICSCs") under development. These applications provide for access to a World Wide Web ("WWW") site, where information pertaining to customer service, such as order status or tips for problem resolution, can be easily obtained. The user of the ICSC uses a WWW browser to search for the requested information. The user may obtain information in the form of text, voice or video. Further, the user may download information to computer data files. Live transfer to a customer service representative is not possible. Examples of patents covering various aspects of communication via data and telecommunication networks are described below.

U.S. Pat. No. 5,533,115 to Hollenbach et al. discloses an advanced ACD for providing information to callers through the Public Switch Telephone Network ("PSTN"). An incoming call from a customer passes through the PSTN to an intelligent peripheral, a service control point, and an ACD. In many cases, the caller is prompted in queue to provide some information, such as the caller's phone number or account number. This information is used to access data records stored in a database which are presented to an agent at an agent terminal. The agent station has access to external communication services, one of which is the Internet. Similarly, U.S. Pat. No. 5,546,452 to Andrews et al. discloses an ACD controlled by a distributed central controller. However, neither Hollenbach et al. nor Andrews et al. appear to disclose an ACD with the capacity to service multimedia callers; only the agent stations have access to the Internet or wide-area networks. Unfortunately, therefore, neither Hollenbach et al. nor Andrews et al. solve the problem of allowing access to a multimedia ACD via data networks. Furthermore, neither Hollenbach et al. nor Andrews et al. teach or disclose the use of an MMACD Server as connection manager for data network callers.

U.S. Pat. No. 5,500,859 to Sharma et al. discloses a personal communications system operating on a personal computer which allows the user to select between different communications functions, including telephone call, voice mail, fax transmissions, and multimedia mail. Sharma et al., however, does not solve the problems of providing a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means and providing an MMACD server to act as connection manager for callers.

U.S. Pat. No. 5,384,771 to Isidoro et al. discloses a multimedia call configuration system. Isidoro et al. uses an object oriented command set to establish a multimedia call over a broadband network. One command set relates to the call configuration and another--to the connection configuration. Unfortunately, however, Isidoro et al. does not solve the problem of providing a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means; rather, it is directed toward establishing communications between multiple specifec parties and has no automatic call direction (ACD) capability at all.

U.S. Pat. No. 5,546,324 to Palmer et al. discloses a video conferencing system used over a data network to communicate among terminals of the network. However, Palmer et al. is only directed to transferring video and audio data. Unfortunately, Palmer et al. does not solve the problems of providing a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means and providing a MMACD server to act as connection manager for callers. Rather, it is directed toward establishing communications between multiple specific parties and has no automatic call direction (ACD) capability at all.

U.S. Pat. No. 5,526,353 to Henley et al. discloses a system and method for communication of audio data over a packet-based network. Henley et al., however, does not solve the problems of providing a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means and providing a MMACD server to act as connection manager for data network callers.

U.S. Pat. No. 5,241,625 to Epard et al. discloses a system for sharing screens over a heterogeneous network. Similarly, U.S. Pat. No. 5,473,680 to Porter discloses methods and apparatus for interfacing with application programs to manage multimedia multiparty communications using different hardware systems and devices. Neither Epard et al. nor Porter solve the problem of providing a multimedia telecommunication ACD system which allows access to the call center via a plurality of access means or a MMACD server acting as a connection manager for callers. Rather, they are directed toward establishing communications between multiple specific parties and have no automatic call direction (ACD) capability at all.

See, United States Patent 6,046,762 (Sonesh, et al., April 4, 2000), expressly incorporated herein by reference.

SUMMARY OF THE INVENTION

The present invention provides an open application programming interface (API) telephony server (OTS™), and a set of new functions particularly enabled thereby.

The API of the OTS™ is provided in the Appendix. Of particular note is that the OTS™ application is present as a dynamic link library (DLL) executing on a Microsoft Windows NT server, and the OTS™ DLL includes functionality to call and pass parameters with multiple instances of multiple other DLLs. Thus, the functionality of the OTS™ is not fixed nor limited to predefined functionality. Further, certain platform dependent code may be segregated, thereby enhancing the hardware independence of the core OTS™ DLL. The particular command is the "RunExtension" command, (and the related "ExtCmpMsg" and "OTSExtensionEntry" commands).

The OTS™ has a corresponding client system resident DLL, which may also have the ability to call external DLLs on the client system.

The present invention also provides enhanced functionality for VOIP systems.

These functions include proactive communications between a telephony server and client systems over the Internet; one click connection to a call center from a Web browser, based on a prestored user profile defining a method of communication and parameters therefore; credit card support for use of VOIP telephony; and an application service provider model for computer telephony (differing from Centrex), eliminating the need for costly dedicated customer premises equipment, while allowing efficient short term usage of adequately configured systems.

It is an object of the invention to provide a system of open telephony that features proactive telephone dialing of the user of a server computer by a representative of the service that is initiated by some action or response of the user to items on the server.

It is another object of the invention to provide a system of proactive dialing that allows the user himself to initiate the dialing by using a one-click function of the system.

It is another object of the invention to provide a system that integrates debit or credit card use with the proactive dialing, allowing efficient payments for telephony usage. Alternately, billing may be by way if an ISP, local telephone carrier, micropayments, prepayment, or the like.

These objects and other objects that will become apparent from the following specification are fulfilled by a telephony system that comprises a server coupled to telephone service and having a program that provides telephone dialing of the user of the server by a representative of the service. Thus, a user of a Web site may be proactively contacted by a call center. This contact may be randomly generated, but preferably, it is initiated based on a pattern and/or status of use of a Web site by the user. This system may also be used to provide voice mail or to contact the user for customer service, although other means are available in that case.

Thus, a call may be placed to a user based on an apparent need for assistance, an indication that a personalized salesperson call would be effective in closing a transaction, or for various other reasons.

In operation, the client system stores a cookie or other persistent state identifier, which allows the user to select contact preferences. For example, the preference may be for a VOIP connection to the same IP address as the Web browser, to a different IP address or telephone number, to a dial-up line (then presently in use by the browser, in which case the call center would wait to call until the line is clear), or the like. The call center may be able to cooperate with an Internet Service Provider to capture the automatic number identification (ANI) of a telephone line, even if the user does not know the number, or is mobile.

The cookie may be placed on the client machine during an initial registration, and persist thereafter. If the cookie is lost, it may be recreated or retrieved from a server based on a username/password.

Thus, only a single pointing device action is necessary to initiate the outbound call process from the call center. It is noted that this process need not involve a call center, and would be operable even between, for example "Chat" partners. By providing a set of preferences to account for various hardware limitations and implement required control sequences, the incompatibilities between the various user hardware are minimized, and the process made "user friendly".

In some cases, the user is required to pay for calls. In that case, the telephony system preferably integrates debit or credit card verification and charging, for example used with the proactive dialing. A micropayment accounting mechanism may be part of the system.

Micropayments are often preferred where the amount of the transaction does not justify the costs of complete financial security. In the micropayment scheme, typically a direct communication between creditor and debtor is not required; rather, the transaction produces a result which eventually results in an economic transfer, but which may remain outstanding subsequent to transfer of the underlying goods or services. The theory underlying this micropayment scheme is that the monetary units are small enough such that risks of failure in transaction closure is relatively insignificant for both parties, but that a user gets few chances to default before credit is withdrawn. On the other hand, the transaction costs of a non-real time transactions of small monetary units are substantially less than those of secure, unlimited or potentially high value, real time verified transactions, allowing and facilitating such types of commerce. Thus, the rights management system may employ applets local to the client system, which communicate with other applets and/or the server and/or a vendor/rights-holder to validate a transaction, at low transactional costs.

The following U.S. Patents, expressly incorporated herein by reference, define aspects of micropayment, digital certificate, and on-line payment systems: 5,930,777 (Barber); 5,857,023 (Demers et al.); 5,815,657 (Williams); 5,793,868 (Micali); 5,717,757 (Micali); 5,666,416 (Micali); 5,677,955 (Doggett et al.); 5,839,119 (Krsul; et al.); 5,915,093 (Berlin et al.); 5,937,394 (Wong, et al.); 5,933,498 (Schneck et al.); 5,903,880 (Biffar); 5,903,651 (Kocher); 5,884,277

(Khosla); 5,960,083 (Micali); 5,963,924 (Williams et al.); 5,996,076 (Rowney et al.); 6,016,484 (Williams et al.); 6,018,724 (Arent); 6,021,202 (Anderson et al.); 6,035,402 (Vaeth et al.); 6,049,786 (Smorodinsky); 6,049,787 (Takahashi, et al.); 6,058,381 (Nelson); 6,061,448 (Smith, et al.); 5,987,132 (Rowney); and 6,061,665 (Bahreman). See also, Rivest and Shamir, "PayWord and MicroMint: Two Simple Micropayment Schemes" (May 7, 1996), expressly incorporated herein by reference; Micro PAYMENT transfer Protocol (MPTP) Version 0.1 (22-Nov-95) et seq, http://www.w3.org/pub/WWW/TR/WD-mptp; Common Markup for web Micropayment Systems, http://www.w3.org/TR/WD-Micropayment-Markup (09-Jun-99); "Distributing Intellectual Property: a Model of Microtransaction Based Upon Metadata and Digital Signatures", Olivia, Maurizio, http://olivia.modlang.denison.edu/~olivia/RFC/09/.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

There are thus four major aspects of the system:

1. The first is the object telephony server (OTS™), a computer program with an application programming interface (hereinafter referred to as ("API"). Preferably, this OTS™ is extensible by the ability to call external DLLs, and further can handle multiple simultaneous tasks by spawning multiple instances of required DLLs. Therefore, a single instance of the OTS™ can handle hundreds of voice channels simultaneously, since the load on the OTS™ is minimal, either due to the intrinsic simplicity of the functions handled, or the externalization of complex or time-consuming tasks.

2. The second is proactive dialing, employing Internet cookies to track users and maintain communications preferences. The server thus understands which users are actively using a site, what their status within the site is, and can make some sort of an intelligent or automated intelligence decision as to who should be contacted for voice communications.

3. The third is a one-click (user initiated) telecommunications function and its ability to invoke a set of user preferences, which are present either in a cookie or referenced on the server via cookie. It initiates telecommunications, by through a selected communications channel (or set of channels or priority protocol). Thus, the functionality on the Web page of the Web site is transparent to the call type, which is handled through the OTS™, rather than the Web server.

Where appropriate, debit or credit card charges may be imposed automatically, or after an authorization.

4. The fourth one is application service provider (hereinafter referred to as ("ASP") model. The architecture of the system of the invention and the functionality of the system potentially allow the creation of a new market for an application service provider for telephony servers. In a fast moving market, people do not want to invest long-term in telephony equipment or telephony servers. The invention provides for capitalizing on the rapid state of change by renting the telephone server software, hosted either externally, while providing minimum on-premises hardware. This model may also be used to implement a "try-it-before-you-buy-it" plan, or an out-sourceing plan. In other words, since Internet bandwidth is relatively cheap, a VOIP system running on a computer network with appropriate audio interfaces would require minimal dedicated hardware, and the telephony server and telephone network interfaces may be hosted remotely. Typically, the customer premises equipment would include a Windows NT server with an appropriate number and type of Dialogic boards, with the OTS™ server executing locally.

Part of this invention includes a new economic model for the sale and use of telephony equipment. On the telephony side, the OTS™ system it is very comprehensive in terms of all functions right in one box, with functions implemented primarily in software, especially at the client systems. An important aspect is the extensibility across multiple sites, across the network and the scaling that results from being able to put multiple systems that behave together as one, the ability to distribute the load among multiple server boxes and still have the system act as one.

The present invention also provides a telephony computer-network appliance, hereinafter called a desk-top box. This desk-top box provides a twisted pair Ethernet interface (or other standard computer network interface), a communications processor, supporting G.711 or G.723 or other standard network audio protocols, an audio interface, e.g., a handset or speakerphone, and a manual user interface, e.g., pushbuttons. Advantageously, but not necessarily, the desk-top box supports the client portion of the OTS™. The desk-top box may also integrate other functions, such as telephone answering device and voicemail/autoattendant, e-mail and Web browsing, and the like.

The OTS™ is a scaleable architecture, and in large installations, is a distributed architecture, which does not require larg amounts of computing power to operate. This is a result of two factors. The first is that this is a Windows NT based system, such as the OTS™ can communicate across a network, and second, much of the telephone call progress processing is done by Dialogic boards, which are hardware level support for telephony applications. The server merely has to control the system, not implement the functions. The software makes it behave in a certain way.

The system provides an API that translates the minutia of hardware level and register level programming of hardware telephony cards, such as the aforementioned Dialogic cards, into a application program level interface so that different versions of the server can operate with different hardware. If a user desires to upgrade his hardware to either a different Dialogic board, even if a change is made in the low level interface, or migrate to some other vendor, that is possible. Thus, while the OTS™ may require certain modifications, the application software will be portable to the new platform.

At high level, the user doesn't have to program in low level functions. Therefore, the user program need only deal with high level functionality in user interface.

Programming of the high level functions can be almost at a scripting level, as opposed to a coding level, although the application programs for the OTS™ are not scripts; some code writing is still required, but it is high level code. In fact, it is possible to provide a secondary product, which supports a scripting function for the OTS™..

The OTS™ DLL has extensibility, or the ability to call from the server a DLL which does not have to be defined at the time the server is coded and finalized. It allows developers to use any language that they want, to script out the logical flow of their calls and the necessary database and business rules and encapsulate it into that DLL which is called by the OTS™ application of the invention, and which can be invoked by the application through a stimulus from a third party application. Thus, these called DLLs may be provided by third parties.

The OTS™ API is a real time asynchronous API, that supports up to over 400 ports meaning, that to control a large number of phone calls, e.g., a system with 100 lines. A client application may be developed which is controlling the server, and communicates through the basic API. However, in this case, it is necessary to develop a multi-threaded application that can moderate the real time asynchronous communication needs of all of those 100 telephone calls simultaneously, which this is complex. Through the preferred DLL mechanism according to the present invention, it is possible to implement the logic required to handle one individual phone call in the form of a DLL, using any programming language desired.

For example, the routing of the call may be: the call comes in, some data is received from the telephone network, that data is matched against an external database, conditions in the call center are analyzed, and then based on this analysis, the routing of the call is determined and it is given an instruction to route this call here or play this message to the caller. That call must be multi-thread into the control application, among other call handling needs, to do it through a conventional API. If this is implemented in a DLL, the primary call control application can simply, when a call comes in, pass it to the DLL, and the DLL is independently invoked simultaneously on as many phone calls that need that DLL at that moment in time. So if 100 calls come in requiring the same logic, as each call comes in, a new instance of the DLL is invoked, and it runs independently from the other instances. What this does is to take advantage of some of the internal capabilities of Windows NT, and eliminate the need to actually code the multi-threading and the management of the multiple calls in the logic itself. Tell it has to call that DLL at this point in time.

Such a called DLL can communicate directly to the hardware layer under NT, but it doesn't in the present invention. However, if it was require to support a custom piece of hardware, it would be possible, at the DLL level, if properly installed under NT, to communicate with the hardware, thus providing an open architecture and intrinsic extensibility.

Instead of somebody having to buy the software and implement the system based on a three or five year amortization, they can lease the software on a monthly or yearly basis, and since telephony is to some degree hardware based, they will have to implement an NT box with dialogic cards. That cannot be outsourced unless one has a Centrex type system.

According to the present invention, the OTS™ server resides in the same server as the telephone hardware, since this server interfaces with the boards. It is possible to split the high and low level functions into different files, but since the OTS™ is a relatively small program, this is generally unnecessary. Thus, on a high level, the system software can be distributed across multiple boxes, in multiple locations.

Software that operates at a higher level than the API is client application software for the DLLs. The DLLs can be run on the same NT box as the software that invokes them, but the OTS™ system is not limited to that configuration. It is also possible to have a slave hardware control layer running on that box, but at the API layer, separated out and running on a different server.

Two advantages are provided with an application services provider mode; the first is that the provider does not give up control over the software, which makes it easier to justify a short term implementation, and the second is that it makes it easier for the provider to suggest to a client that they can easily have distributed hardware. Because their software no longer has to be ported over multiple platforms, they can have one instance of their application software, and the provider's API to hardware abstraction layer software can implement the cross platform or the distributed architecture.

As an application service provider, new revenue models are available for use of the system. Thus, it is possible to arrange a monthly lease, pay by the minute, or pay by the call, or some combination of these. It is also possible to have payment by the number of minutes or seconds that certain types of resources in the system are used. It is noted that, while charging by the minute for the use of telecommunications services is the general revenue model of telecommunications carriers, this is not a typical model for software providers or telephone hardware providers. It is also possible to isolate out voice processing resources, conferencing resources, IP resources, switching resources, line interfaces, voice recognition, text to speech, or other similar resources, either isolating out features or isolating out physical resources and metering them on a microscale.

Another advantage of an application service provider it is also possible for the central network to actually interface with the telephone company, and to have some or all of the telecommunications hardware at the provider's site, using packet switched networks to communicate between the telecommunications carrier and the call center. It is thus possible to provide a service that requires little dedicated hardware at a client's site, with a VOIP implementation.

A user can subscribe to the provider's service and use VOIP, and that would implement all of the call center functions and the network connections. This technology therefore enables virtual call centers. The minimum bandwidth that a user would need is about 9600 baud, although 128k (2B) (DSL or ADSL) or 144k (2B + D) (ISDN), per operator, would be superior, depending on desired data communications bandwidth and voice quality.

"One-click" is the business model that allows somebody, while they're browsing the web or otherwise in a data communications mode, to have a single action that triggers a sequence of events that ultimately leads to voice communication between that person and a call center, or that person and another person in the event of a one-to-one communication scheme. The different options are a dial back or the user can initiate the call. The call center can initiate the call to the user, whether it is over IP, the same phone line the user is using now, or a different phone line.

The present invention, however, does not require a consistent transport protocol for all users, e.g., VOIP. The invention provides that only a single click to establish the preferred means of communication as opposed to having different clicks depending on a preferred means of communication. The idea is that if the user is on a dialup line for their internet connect, and they would prefer to speak on a separate telephone, that is different from VOIP over the existing connection. While VOIP is probably the most trivial in broadband scenarios, it poses issues for many consumers, especially those using dial-up connections. On a dialup line, there is basically no bandwidth left after implementing the VOIP, which has inferior quality to analog voice over the same line.

In peer-to-peer communications, both the requester of the communication and the recipient of the communication may each have preferences. Therefore, the present invention accommodates the preferences of each party, so long as they are possible and compatible. Even with communications to a call center, the user preferences may define a Spanish language operator or optimally route the call based on a user profile.

Existing VOIP applications may give the user the same kind of functionality through use of the ANI/DNIS. If H.323 is used, the data communications components may be used to allow the two systems on either end to know something about each other.

For example, the present one-click process (generally peer-to-peer) could be implemented on a web site like e-Bay. If somebody wants to communicate with an auctioneer, he pushes an icon button, which then serves to establish an IP communication session on the requesting user's side. On the auctioneer's side, however, the call be directed to ring his cell phone. Thus, there is no a priori specification as to what the mode of communication on either end of the connection is.

It is noted that the HTML code necessary to place the functionality on a web page is small, since the hyperlink only access a URL server, which retrieves the cookie and possibly other information about the user and his connection. Thus, it can be liberally distributed to a variety of Web sites.

One of the advantages of this over just a phone number is that it allows a degree of anonymity, especially in VOIP to VOIP communications. The user preferences for connection or callback may include a number of parameters, such as rules based on who is trying to contact one, what time of day, day of week, etc., and what one's preferred method of connection is. During off hours, one can direct communications to a phone answering machine or go into voicemail. With voicemail and full interactive voicemail, for example, one can provide code words for different users.

For example, users connecting to the service provider without a password would be presented with a schedule of dates and times of business. Users connecting with a password the service provider can have different passwords for different people. If the service provider wants to remove certain people from his password list, he can remove people individually, which would not be possible with a global password. This is simple to implement, but it is a service that nobody is currently providing.

The service provider could provide the user with the ability to use credit or debit cards or provide services on a pre-paid basis. Micropayment technology can be used. The service may be wholly or partially advertiser supported. In one embodiment, billing is entrusted to the ISPs. Advertising may therefore accompany the telephone call, either visually (static or multimedia) or audibly through a browser, or auditorily during voice communications. A combination of streaming audio ads and maybe spring ads may be presented to the user as he sets up the connection or during the connection, at least on one side.

The service provider can open up another window so that whatever website the user is on can stay open, and the service provider can open up a window that it knows its server controls during the connection process. The service provider can inform people of the status of the connection trying contact method number one or whatever and the service provider can put an ad in there. Or the service provider can give the person who is subscribing to the service the option to provide their own advertising. They can pay the service provider on a use basis, but the subscriber can put certain advertising files in there to be shown to people while they are waiting to be connected or while they are connected.

Since VOIP calling is essentially no cost, the facilitation of its use basically provides a way for people to have their own toll free number without an owning 800 number. Even if the subscriber had to pay to reserve bandwidth on an IP network, his costs are going to be less than a penny a minute.

Alternative means of contact through this conventional phone network may be provided at additional costs. It is also possible to bill right to a phone bill

The service provider can bill a specific amount per month and bundle a certain amount of usage. It can be like a cellular plan, flat amount of money per month, up to so many minutes that can't be carried forward. If you go over your minute usage, the subscriber would be charged for additional minutes. It is not desirable to bill micropayments on credit cards or phone bills. It is better to reduce those transactions, but pre-paid accounts or flat fees are more desirable.

According to an outbound proactive calling embodiment of the invention, a preferred trigger for a call is a shopping cart status. Just as a sales clerk assists customers at checkout in a brick-and-mortar store, so can an on-line sales associate assist a user in selecting and purchasing items of interest. The receipt of an incoming voice communication by a user may be automatic or permissive. In the permissive case, a dialog box is called up to request permission to conduct a voice communication with the user. In some instances, displaying a text chat window will be as effective, and therefore such a function is preferably supported by the present invention as one alternative communication means. In some instances, playing a sound file will be as effective, and therefore such a function is preferably supported by the present invention as one alternative communication means.

According to the invention, telephony hardware per se is not required, except in certain circumstances. The telephony hardware is necessary, for example, in a gateway application, where there is a need to convert between IP and circuit based voice, or where one wants to engage the IP participant into other conventional telephony functions, that are provided by services of the Dialogic boards or equivalent, like conferencing or recording, and if the communication is not pure IP, or one wants to use Dialogic board resources to analyze the progress of a call to bring that call into the gateway itself. It is noted that, for VOIP, a VOIP gateway is required to interface with the Dialogic boards.

Another advantage of the present invention is that, by allowing virtual call centers, it allows operators to become independent contractors and call centers to staff at lower levels, since excess calls may be transferred outside a facility and handled by overflow or freelance operators of appropriate training and experience.

Another aspect of the present invention provides a remote control applet, allowing a remote operator co control (to co-control) the screen interface of a user's computer. This applet could also record actions and/or block functions, for example not allowing the agent to "click", e.g., blocking the MouseDown event, or tagging certain page elements as being local-execute only. Alternately, all actions could require a client-side confirm.

Many aspects of the present invention employ known techniques, although employed in different contexts herein. Therefore, it is understood that these known techniques and those associated with them may be employed in conjunction with the present invention, to the extent consistent therewith.

It is also understood that the various aspects of the invention may be employed together, individually or in subcombination. Further, it is understood that the present techniques are not limited to use on the Internet, as presently known, and may be applied to a large number of human computer interface systems.

While the above detailed description has shown, described and pointed out the fundamental novel features of the invention as applied to various embodiments, it will be understood that various omissions and substitutions and changes in the form and details of the system and method illustrated may be made by those skilled in the art, without departing from the spirit of the invention. Consequently, the full scope of the invention should be ascertained by the appended

CLAIMS:

What is claimed is:

1. An internet telephony system comprising a browser display having a hyperlink, said hyperlink communicating with a server, retrieving a user's preferences, and initiating a voice communication with the user, through a communications channel defined by the user preferences.

2. The telephony system according to claim 1, wherein said user preferences define a communications mode selected from the group consisting of voice over IP and analog voice.

3. The telephony system according to claim 1, wherein the user's preferences are retrieved in a cookie.

4. The telephony system according to claim 1, wherein a cookie identifies a user.

5. The telephony system according to claim 1, wherein the user, in order to communicate with the server to request a voice communication, has only a single essential action.

6. An Internet telephony system comprising a client system having an Internet browser, and a server hosting a Web site, wherein a message is transmitted from the server to the client system based on the user's status with respect to Web site, the seeking to establish a voice communication session.

7. The telephony system according to claim 6, wherein a set of user preferences are retrieved by the server, defining a preferred a communications mode selected from the group consisting of voice over IP and analog voice.

8. The telephony system according to claim 7, wherein the user's preferences are retrieved in a cookie.

9. The telephony system according to claim 7, wherein a cookie identifies a user.

10. The telephony system according to claim 7, wherein the user, in order to communicate with the server to request a voice communication, the user's status with respect to the web site comprises an analysis of a shopping cart.

11. An Internet telephone system, comprising a browser display having a hyperlink, said hyperlink communicating with a server, retrieving a user-related data, and initiating a voice communication with the user, wherein the user is charged for the communication.

12. The telephony system according to claim 11, wherein said voice communication is selected from the group consisting of voice over IP and analog voice.

13. The telephony system according to claim 11, wherein the user charge is a micropayment.

14. The telephony system according to claim 11, wherein a set of user preferences is retrieved in a cookie from the browser.

15. The telephony system according to claim 11, wherein a cookie identifies the user.

16. The telephony system according to claim 11, wherein the user, in order to communicate with the server to request a voice communication, has only a single essential action.

17. A telephony server, comprising an application program communicating directly with telephony hardware to implement telephony system control, and an application programming interface, wherein said application program includes as one of its is application programming interface functions a call to an external program.

18. The telephony server according to claim 17, wherein the application program is a dynamic link library adapted to run under Microsoft Windows.

19. The telephony server according to claim 17, wherein the application program may spawn a plurality of instances of the external program simultaneously.

20. The telephony server according to claim 17, wherein the application program has a component running on a telephony server and a component running on each telephony client.

21. The telephony server according to claim 17, further comprising a communications link to a Web server, for coordinating telephony functions and Web server functions.

ABSTRACT

An internet telephony system comprising a browser display having a hyperlink, said hyperlink communicating with a server, retrieving a user's preferences, and initiating a voice communication with the user, through a communications channel defined by the user preferences. Also provided is an Internet telephony system comprising a client system having an Internet browser, and a server hosting a Web site, wherein a message is transmitted from the server to the client system based on the user's status with respect to Web site, the seeking to establish a voice communication session. A telephony server, comprising an application program communicating directly with telephony hardware, and an application programming interface, wherein said application program includes as one of its is application programming interface functions a call to an external program. An Internet telephone system, comprising a browser display having a hyperlink, said hyperlink communicating with a server, retrieving a user-related data, and initiating a voice communication with the user, wherein the user is charged for the communication.

DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION

Attorney Docket No.
BUFFALO 201

As the below named inventors, I/We hereby declare that:

My/Our name(s), residence(s), post office address(es) and citizenship(s) is/are as stated below next to my/our name(s).

If one name appears below, I am the sole inventor of the subject matter sought to be patented.

If two or more names appear below, we are joint inventors of the subject matter sought to be patented.

I/We believe I/We am/are the original; and first inventor(s) of the subject matter which is claimed and for which a patent is sought on the invention entitled

VOICE OVER INTERNET CALL CENTER INTEGRATION

the specification of which
[x] is attached hereto.
[ ] was filed on ______________ as application Serial No. _________.

I/We hereby state that I/We reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I /We acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56(a).

I/We also acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.63(d), which occurred between the filing date of the prior application and the filing date of the continuation-in-part application, if this is a continuation-in-part application.

I/We hereby claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for the patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application: __________ Application No.
filed
Priority Claimed: ____ Yes ___ No

I/We hereby claim the benefit under Title 35, United States Code, Section 119(e) or 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial No.	Filing Date	Status (patented, pending, abandoned)
Application Serial No.	Filing Date	Status (patented, pending, abandoned)

I/We hereby declare that all statements made herein of my/our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I/We hereby appoint the following attorneys and/or agents to represent me with respect to the above identified U.S. Patent Application, and to prosecute any continuations, continuations-in-part, reissue applications and/or reeaxaminations with respect to these applications and to transact all business in the Patent and Trademark Office connected therewith, and hereby expressly revoke all prior powers, whatever they may be, heretofore had herein:

> Karl F. Milde, Jr., Reg. No. 24, 822; Steven M. Hoffberg, Reg. No. 33,511 and Kenneth E. Macklin, Reg. No. 20,875, all of 10 Bank Street, Suite 460, White Plains, New York 10606, my/our attorneys with full power of substitution and revocation.

Please address all telephone calls to Steven M. Hoffberg, Esq. at telephone No. (914) 949-3100.

Please address all correspondence to:

Steven M. Hoffberg, Esq.
MILDE, HOFFBERG & MACKLIN, LLP
10 Bank Street - Suite 460
White Plains, New York 10606

David Friedman
NAME OF INVENTOR

INVENTOR'S SIGNATURE

5/26/00
DATE

400 Columbus Avenue, Valhalla, NY 10595
RESIDENCE

USA
CITIZENSHIP

Same as above
POST OFFICE ADDRESS

Wai Wu
NAME OF INVENTOR

INVENTOR'S SIGNATURE

5/26/00
DATE

400 Columbus Avenue, Valhalla, NY 10595
RESIDENCE

USA
CITIZENSHIP

Same as above
POST OFFICE ADDRESS

APPENDIX

Control API

AddDRecord

Adds a dialing record to an existing project

Syntax

AddDRecord(**pID** as integer, **d** as string, **len** as integer) as integer

Parameter	**Description**
pID	Project number
d	Dialing record (max length of 255 bytes)
len	Length of the record

AddPriRecord

Adds the dialing record to the front of the dialing record queue

Syntax

AddPriRecord(**pID** as integer, **Record** as string, **length** as integer) as integer

Parameter	**Description**
pID	Project number
Record	Dialing record
Length	Length of the dialing record

AddProject

Creates a project in OTS ™

Syntax

AddProject(**pID** as integer) as integer

Parameter	**Description**
pID	Project identification number

ClearDTMF

Clears the hardware DTMF buffer

Syntax

ClearDTMF(**pID** As Integer, **CallID** as integer) As Integer

Parameter	**Description**
pID	Project Number
CallID	Call identifier

Connect

Connects the application with the OTS ™ through TCP/IP
*Note: Winsock2 required

Syntax

Connect(**name** as string, **port** as integer) as Boolean

Parameter	Description
name	Server Name
port	Port Number

DeleteRecord *

Deletes record by the record ID

Syntax

DeleteRecord(**pID** as integer, **RecordID** as string, **Length** as integer) as integer

Parameter	Description
pID	Project number
RecordID	Record identifier number
Length	Length of the record

DialDigit

Dials DTMF digits on an existing call.

Syntax

DialDigit(**pID** As Integer, **CSN** As Integer, **CallID** As Integer, **DigitString** As String) As Integer

Parameter	Description
pID	Project Number
CSN	Command Sequence Number
CallID	Call identifier
DigitString	Dial digits numbers

DisableEvent

Disables call status, and number request events. Writes call status to designated file

Syntax

DisableEvent(**pID** As Integer, **RFileName** As String) As Integer

Parameter	Description
pID	Project Number
RfileName	File Name

DisableOutBound

Disables the outbound calls

Syntax

DisableOutBound(**pID** as integer) as integer

Parameter	**Description**
pID	Project number

EnableEvent

This is the default setting of OTS ™

Syntax

EnableEvent(**pID** As Integer) As Integer

Parameter	**Description**
pID	Project Number

EnableOutBound

Enables the outbound calls

Syntax

EnableOutBound(**pID** as integer) as integer

Parameter	**Description**
pID	Project number

FlashDialingQueue

Deletes all records in the dialing record queue

Syntax

FlashDialingQueue(**pID** as integer) as integer

Parameter	**Description**
pID	Project number

GetCallIDByLine

Obtains the CallID from a line number

Syntax

GetCallIDByLine(**pID** As Integer, **CSN** As Integer, **LineNumber** As Integer) As Integer

Parameter	Description
pID	Project number
CSN	Command Sequence Number
LineNumber	Number of the particular line

GetCallInfo

Gets the call information

Syntax

GetCallInfo(**pID** as integer) as integer

Parameter	Description
pID	Project number

GetDTMFs

Gets DTMF digits

Syntax

GetDTMFs(**pID** As Integer, **CSN** As Integer, **CallID** As Integer, **TermDigitASCII** As Integer, **Digits** As Integer, **Timeout** As Integer) As Integer

Parameter	Description
pID	Project Number
CSN	Command Sequence Number
CallID	Call identifier
TermDigitASCII	Terminating digit (in ASCII)
Digits	Max number of digits
Timeout	Max time allowed

GetLineByCallID

Obtains the line number from the CallID

Syntax

GetLineByCallID(**pID** as integer, **CSN** as integer, **CallID** as integer) as integer

Parameter	Description
pID	Project number
CSN	Command Sequence Number
CallID	Call Identifier

Table of Contents

GetMsg

Retrieves an unsolicited message stored in the OCX control message queue.

Syntax
GetMsg(**string** as string, **len** as integer) as integer

Parameter	**Description**
string	A place holder for the message
len	Length of place holder

Hangup

Hangs up an existing call.

Syntax
HangUp(**pID** As Integer, **CSN** As Integer, **CallID** As Integer) As Integer

Parameter	**Description**
pID	Project Number
CSN	Command Sequence Number
CallID	Call identifier

Init

Initializes application TCP stack

Syntax
Init()

Returns:	True – Success
	False – Fail

KillConnection

Kills the connection of the agent from OTS ™

Syntax
KillConnection(**pID** As Integer, **CSN** As Integer, **name** As String) As Integer

Parameter	**Description**
pID	Project number
CSN	Command Sequence Number
Name	Agent Name

PlayMsg

Plays a voice file

Syntax

PlayMsg(**pID** As Integer, **CSN** As Integer, **CallID** As Integer, **MsgFile** As String, **TermDigitASCII** As Integer, **Digits** As Integer) As Integer

Parameter	**Description**
pID	Project Number
CSN	Command Sequence Number
CallID	Call identifier
MsgFile	The voice file name and path
TermDigitASCII	Terminating digit (in ASCII)
Digits	Max number of digits

ProcessMsg

Translates an unsolicited message to an ActiveX event

Syntax

ProcessMsg(**MsgStr** as string)

Parameter	**Description**
MsgStr	The actual message

RecordMsg

Records a voice file

Syntax

RecordMsg(**pID** As Integer, **CSN** As Integer, **CallID** As Integer, **MsgFile** As String, **TermDigitASCII** As Integer, **Digits** As Integer, **MaxSeconds** As Integer, **MaxSil** As Integer, **Optionals** As Integer) As Integer

Parameter	**Description**
pID	Project Number
CSN	Command Sequence Number
CallID	Call identifier
MsgFile	The voice file name and path
TermDigitASCII	Terminating digit (in ASCII)
Digits	Max number of digits
Timeout	Max time allowed
MaxSeconds	Max time allowed to recorded the voice file
MaxSil	Max time allowed to be silenced
Optionals	

RemoveProject

Removes a project from the OTS ™

Syntax
RemoveProject(**pID** As Integer) As Integer

Parameter	**Description**
pID	Project Number

Restartproject

Restarts a stopped project

Syntax
Restartproject(**pID** as integer) as integer

Parameter	**Description**
pID	Project number

RouteCall

Routes a call to the next available agent of the project

Syntax
RouteCall(**TargetProjectID** As Integer, **CSN** As Integer, **CallID** As Integer, **Timeout** As Integer) As Integer

Parameter	**Description**
TargetProjectID	Project Number
CSN	Command Sequence Number
CallID	Call identifier
Timeout	Maximum time allowed

RouteCallEx * (Waiting to be documented)

Syntax
RouteCallEx(**TargetProjectID** as integer, **CSN** as integer, **CallID** as integer, **TimeOut** as integer, **QueueTimePosition** as long, **SkillID** as integer) as integer

Parameter	**Description**
pID	Project number
CSN	Command Sequence Number
CallID	Call identifier
TimeOut	Maximum time allowed to wait

QueueTimePosition
SkillID Capability

RunExtension

Runs the OTS ™ extension DLL

Syntax

RunExtension(**pID** as integer, **CSN** as integer, **CallID** as integer, **ExtClassName** as string,) as integer

Parameter	**Description**
pID	Project number
CSN	Command Sequence Number
CallID	Call identifier
ExtClassName	The name of the class (Must be in ActiveX.DLL format)

Returns:	0 – Success 1 – Fail

SendMsgToAgent

Sends message to the agent from the control

Syntax

SendMsgToAgent(**pID** As Integer, **CSN** As Integer, **AgentID** As String, **Message** As String, **Length** As Integer) As Integer

Parameter	**Description**
pID	Project number
CSN	Command Sequence Number
AgentID	Agent name
Message	Content of the message
Length	Length of the message

SetCallData

Sets associated data with the call

Syntax

SetCallData(**pID** as integer, **CallID** as integer, **CallData** as string, **length** as integer) as integer

Parameter	Description
pID	Project number
CallID	Call identifier
CallData	Call data to be associated with the call
Length	Length of the call data

SetCPA

Sets up the call progress analysis parameter

Syntax

SetCPA(**pID** as integer, **PerfCall** as integer, **StDelay** as integer, **RingCount** as integer, **HelloEdge** as integer, **intflg** as integer, **AnsrDgl** as integer, **MaxAnsr** as integer) as integer

Parameter	Description
pID	The identification number of the project
PerfCall	Turn on perfect call if non zero
StDelay	Start delay is the delay after the dialing has been completed and before starting analysis for Cadence detection, Frequency detection, and Positive Voice detection. 10 ms as default
RingCount	Number of the ring count before detected no answer
HelloEdge	The point at which a connect will be returned to the application 1. Rising Edge (immediately when a connect is detected) 2. Falling Edge (after the end of the salutation)
Intflg	Intercept Mode Flag: This parameter enable or disables SIT frequency Detection, Positive Voice Detection (PVD), and/or Positive Answering Machine Detection (PAMD), and select the mode of operation for Frequency Dectection(CA) DX_OPTEN: Enable Frequency Detection and wait for detection of a connect using Cadence Detection or Loop Current Detection before returning an intercept DX_OPTDIS: Disable Frequency Detection and PVD DX_OPTNOCON: Enable Frequency Detection returns an intercept immediately after detecting a valid frequency DX_PVDENABLE: Enable PVD

DX_PVDOPTEN:	Enable PVD and DX_OPTEN
DX_PVDOPTNOCON:	Enable PVD and DX_ OPTNOCON
DX_PAMDENABLE:	Enable PAMD
DX_PAMDOPTEN	Enable PAMD AND DX_OPTEN

AnsrDgl	Maximum silence period allowed between words in a salutation.
MaxAnsr	Maximum allowable length of answer size.

SetDeviceVol *

Sets volume for the conference card

Syntax

SetDeviceVol(**vDigital** as integer, **vAnalog** as integer, **vStation** as integer) as integer

Parameter	**Description**
vDigital	Volume for the digital card
vAnalog	Volume for the analog card
vStation	Volume for the MSI card

SetProjectLine

Assigns a physical line to an existing project

Syntax

SetProjectline(**pID** as integer, **line** as integer) as integer

Parameter	**Description**
pID	Project identification number
Line	Line number

SetProjectParam

Sets the parameters for a registered project

Syntax

SetprojectParam(**pID** as integer, **DRL** as integer, **PO** as integer, **PL** as integer, **RO** as integer, **RL** as integer, **PM** as integer, **LD** as integer) as integer

Parameter	**Description**
pID	The identification number of the project
DRL	The length of the dialing record (max. 255 bytes)
PO	Offset of the phone number into a record

PL	Phone number length (max. 30 bytes)
RO	Offset of record identifier in a record
RL	Record identifier length (max. 30 bytes)
PM	Pacing Multiplier
LD	Number of second to search for a free agent before OTS ™ drops a connected call

SetProjParam2

Sets the parameters for a registered project

Syntax

SetProjParam2(**pID** As Integer, **DialingRecordLength** As Integer, **PhoneOffset** As Integer, **PhoneLength** As Integer, **RecordIDOffset** As Integer, **RecordLength** As Integer, **PacingMulti** As Integer, **DropTime** As Integer, **DisableAGP** As Integer, **HandleAM** As Integer, **HandleLiveCall** As Integer) As Integer

Parameter	**Description**
pID	The identification number of the project
DialingRecordLength	The length of the dialing record (max. 255 bytes)
PhoneOffset	Offset of the phone number into a record
PhoneLength	Phone number length (max. 30 bytes)
RecordIDOffset	Offset of record identifier in a record
RecordLength	Record identifier length (max. 30 bytes)
PacingMulti	Pacing Multiplier
DropTime	Number of second to search for a free agent before OTS ™ drops a connected call
DisableAGP	Disable agent prediction
HandleAM	Max second to wait for the answer machine greeting finished
HandleLiveCall	0- On 1-Off for detecting live voice

SetWinTarget

Registers the control OCX with the window handle, so that when an unsolicited message arrives a message of **ClientMsg** will be posted to the window's queue specified by **hWnd**

Syntax

SetWinTarget(**hWnd** as Long, **ClientMsg** as Long)

Parameter	**Description**
HWnd	Window handle
ClientMsg	User define windows message

ShutDown

Shutdowns TCP stack and breaks connection with OTS ™

Syntax

ShutDown()

StartESM

Registers the application as a control application with this particular control ID

Syntax

StartESM(**name** as string) as integer

Parameter	**Description**
name	The name of the control ID (any unique name)
	1 - Fail

StopProject

Stops a running project

Syntax

StopProject(**pID** as integer) as integer

Parameter	**Description**
pId	Project number

Control Events

AgentMsg

The message sent by the active agent to the control

Syntax

AgentMsg (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **AgentName** As String, ByVal **Messagedata** As String)

Parameter	**Description**
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
AgentName	Agent's name
Messagedata	Message sent from the active agent

CallDisconnected

The function will be called when a called party disconnects

Syntax

CallDisconnected (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer)

Parameter	Description
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier

CallHookFlashed

Generated when a hook flash is deleted on the line

Syntax

CallHookFlashed (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer)

Parameter	Description
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier

CallInformation

This event is generated when a call is make to Callcallinformation

Syntax

CallInformation(ByVal **Attempts** As Integer, ByVal **Connects** As Integer, ByVal **Drops** As Integer, ByVal **Remains** As Integer)

Parameter	Description
Attempts	Number of calls dial for that particular number in that project
Connects	Call connect (group calls connect)
Drops	Number of calls has been dropped in that particular project
Remains	Number of the dialing number that are remaining in the queue

CallMsg

Will be triggered when one incoming call arrives

Syntax

CallMsg (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer, ByVal **CallData** As String)

Parameter	Description
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier
CallData	Call data to be associated with the call

CallStatus

Detects the status of the arriving call

Syntax

CallStatus (ByVal **EventType** As Integer, ByVal **systime** As Long, ByVal **CallStatus** As Integer, ByVal **Outline** As Integer, ByVal **AgentLine** As Integer, ByVal **RecordID** As String, ByVal **ProjectNumber** As Integer)

Parameter	Description
EventType	< void this >
Systime	System time
CallStatus	Call status type
	N – No Answer
	B – Busy
	O – Operator Intercept
	D – Dropped (no agent)
	M – Answer machine
	T – Transfer
	L – No dial tone
	F – Fax machine/modem
OutLine	Outgoing line
AgentLine	Agent extension number
RecordID	Record number contain from SetProejctParam()
ProjectNumber	Project number

CallStatusChanged

When the status of a call changed

Syntax

CallStatusChanged (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **systime** As Long, ByVal **ChangedType** As Integer, ByVal **CallLineNumber** As Integer, ByVal **AgentLineNumber** As Integer, ByVal **RecordID** As String, ByVal **CauseValue** As Integer, ByVal **CallID** As Integer, ByVal **ExtraInfo** As String)

Parameter	**Description**
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
Systime	System time (number of seconds since 1970)
ChangedType	Call status type N – No Answer B – Busy O – Operator Intercept D – Dropped (no agent) M – Answer machine T – Transfer L – No dial tone F – Fax machine/modem H – Hold P – Pick up call I – Manual outbound call J – Joint conference call K – Hang up
CallLineNumber	Line number of the call
AgentLineNumber	Line number of the agent
RecordID	Record identification number
CauseValue	Only apply to ISDN, and probably country specific
CallID	Call identifier
ExtraInfo	Reserved for Future Use

ComCmpMsg (Command complete message)

This event will be generated when one of the following IVR commands is completed

GetDTMFs
PlayMsg
RecordMsg
DialDigit
Hangup

Syntax

ComCmpMsg (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer, ByVal **rc** As Integer, ByVal **dtmfs** As String)

Parameter	**Description**		
EventHeader	< void this >		
EventType	< void this >		
ProjectNumber	Project number		
rc	Completion reason	Bit Map	
	TM_NORMTERM	0x00000	Normal Termaination
	TM_MAXDTMF	0x00001	Max # of Digits recorded
	TM_MAXSIL	0x00002	Max Silence
	TM_MAXNOSIL	0x00004	Max Non-Silence
	TM_LCOFF	0x00008	Loop Current off
	TM_IDDTIME	0x00010	Inter Digit Delay
	TM_MAXTIME	0x00020	Max Function Time Exceed
	TM_DIGIT	0x00040	Digit mask or Digit Type Term
	TM_PATTERN	0x00080	Pattern Match Silence Off
	TM_USRSTOP	0x00100	Function Stopped by User
	TM_EOD	0x00200	End of Data Reached on Playback
	TM_TONE	0x02000	Tone On/Off Termination
CallID	Call identifier		
dtmfs	DTMF digits (only apply to GetDTMF command)		

ExtCmpMsg

OTS ™ Extension is completed

Syntax

ExtCmpMsg(ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer, ByVal **ExtensionReturn** As Long)

Parameter	**Description**
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier
ExtensionReturn	custom return code from the extension

LineAvailable

Generated when a line becomes available or is assigned to another project.

Syntax

RouteTimeOut (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **Generic** As Integer)

Parameter	Description
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
Generic	

ReflectedCall

Generated when a call is answered and either HandleLiveCall or HandleAM is none zero.

Syntax

ReflectedCall(ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer, ByVal **CallData** As String, ByVal **ConnectType** As Integer)

Parameter	Description
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier
CallData	Call data to be associated with the call
ConnectType	Type of destination 13 – Answer Machine 14 – Live Connect

RequestNumber

Will be generated when OTS ™ outbound dial queue is low (< 20)

Syntax

RequestNumber (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **Generic** As Integer)

Parameter	Description
EventHeader	< void this >
EventType	< void this >

ProjectNumber	Project number
Generic	Generic data field

RouteTimeOut

Generated when the route call command is timed out. In other words, no agent has picked up the call that was previously routed within the specified time in the route call command.

Syntax

RouteTimeOut (ByVal **EventHeader** As Integer, ByVal **EventType** As Integer, ByVal **ProjectNumber** As Integer, ByVal **CallID** As Integer)

Parameter	**Description**
EventHeader	< void this >
EventType	< void this >
ProjectNumber	Project number
CallID	Call identifier

OTSExtensionEntry

Accesses the extension addons functions

Syntax

OTSExtensionEntry(**OTSHook** as Object) as Long

Parameter	**Description**
OTSHook	Access all the extension functions

Methods available in the IDispatch parameter. A –1 returning from any of the following functions indicates a disconnection of the call.

ClearDTMFs

Clears internal digit buffer

Syntax

ClearDTMFs() as long

Dial

Generates dial tone and dial the number

Syntax
Dial(Number as string) as long

Parameter	**Description**
Number	Digits string(max length of 64 bytes)

GetCallData

Retrieves the associated data with the call

Syntax
SetCallData(**Data** as string, **length** as integer) as integer

Parameter	**Description**
Data	Associated data
Length	Length of the associated data

GetDTMFs

Gets phone digits input

Syntax
GetDTMFs(OutDTMFs as string, **term** as integer, **digits** as integer, **maxtime** as integer) as long

Parameter	**Description**
OutDTMFs	Phone digit input
Term	Termination digit (Must be ASCII Character)
digits	Numbers of the digit from the phone input
maxtime	Maximum time allowed

Play

Plays a voice file

Syntax
Play(Filename as string, **term** as integer, **digits** as integer) as long

Parameter	**Description**
Filename	Voice file name
Term	Termination digit (Must be ASCII Character)
Digits	Numbers of the termination digit

Record

Records a voice file

Syntax
Record(Filename as string, **term** as integer, **digits** as integer, **maxtime** as integer, **maxsil** as integer, **tone** as integer) as long

Parameter	**Description**
Filename	Voice file name
Term	Termination digit (Must be ASCII Character)
Digits	Numbers of the termination digit
Maxtime	Maximum time allowed
MaxSil	Maximum silence allowed
Tone	0 – set to record the message without the prefix of beep sound 2 – set to record the message with the prefix of the beep sound

Route *

Route call through agent extension

Syntax
Route(**pID** as integer, **MaxTime** as integer) as long

Parameter	**Description**
pID	Project number
MaxTime	Maximum time allowed

SetCallData

Sets associated data with the call

Syntax
SetCallData(**Data** as string, **length** as integer) as long

Parameter	**Description**
Data	Associated data
Length	Length of the associated data

Client API

Connect

Connects the client application to the OTS ™

Syntax
Connect(**name** as string, **port** as integer) as boolean

Parameter	**Description**
name	The server name
port	Port ID of the server

ConnectCall

Connects two held calls (Agent does not involve with the calls)

Syntax

ConnectCall(**CallID1** as integer, **CallID2** as integer) as integer

Parameter	**Description**
CallID1	Call identifier number one
CallID2	Call identifier number two

CXSendDigits *

Sends digits through current call

Syntax

CXSendDigits(**AgentID** as string, **Digits** as string) as integer

Parameter	**Description**
AgentID	Agent identifier number
Digits	Phone number

DelConference

Destroys the conference created previously by MakeConference

Syntax

DelConference (**CallID1** as integer, **CallID2** as integer) as integer

Parameter	**Description**
CallID1	Call identifier number one
CallID2	Call identifier number two

Dial

Tells OTS ™ to dial a number from one of the lines assigned in the project

Syntax

Dial(**number** as string) as integer

Parameter	**Description**
number	Phone number to dial out

DialLocal

Dials a number from the line the agent application specified with SetUserData

Syntax

DialLocal (**number** as string) as integer

Parameter	Description
number	Phone number

Disconnect

Disconnects TCP connection with server

Syntax
Disconnect()

EnableExMsg

Enable the reception of unsolicited messages

Syntax
EnableExMsg()

GetAgentCallID

Gets agent call identifier

Syntax
GetAgentCallID(**AgentName** as string, **CallID** as integer) as integer

Parameter	Description
AgentName	Agent name
CallID	Phone call identification number

GetMsg

Retrieves a message from the Client OCX control
Syntax
GetMsg(**string** as string, **len** as integer) as integer

Parameter	Description
string	The actual message from the OTS ™
len	The length of the message

HangUp

Terminates the current call.

Syntax
HangUp()

HangupLocal

Hangs up the line that the agent application specified from the SetUserData

Syntax
HangUpLocal()

Hold

Puts the current call on hold

Syntax
Hold(**cId** as integer) as integer

Parameter	**Description**
cId	Call identifier

Init

It initializes the client application with the OTS ™

Syntax
Init()

MakeConference

Creates three ways conference call with the agent

Syntax
MakeConference(**CallD1** As Integer, **CallD2** As Integer) As Integer

Parameter	**Description**
CallID1	Call identifier One
CallID 2	Call identifier Two

NotReady

Put agent into a not ready state

Syntax
NotReady()

ProcessMsg

Translates an unsolicited message to a ActiveX event

Syntax
ProcessMsg(**MsgStr** as string)

Parameter	**Description**
MsgStr	The actual message

Ready

Tells OTS ™ that the agent is ready to receive a call from the designated group

Syntax

Ready()

Retrieve

Retrieves the call that has been put on hold

Syntax

Retrieve(**cID** as integer) as integer

Parameter	Description
cID	Call Identifier for the call on hold

SendMsgToCtrl

Sends a message to the control application

Syntax

SendMsgToCtrl(**MessageData** As String, **length** As Integer) As Integer

SetCallData *

Sets associated data with the call from the agent

Syntax

SetCallData(**CallData** as string, **Length** as integer) as integer

Parameter	Description
CallData	Associated data
Length	Length of associated data

SetAgentAuxCap * (Waiting to be documented)

Syntax

SetAgentAuxCap(pID As Integer, SkillMask1 As Long, SkillMask2 As Long) As Integer

Parameter	Description
pID	Project number
SkillMask1	
SkillMask2	

SetUserData

Establishes connection as agent login

*Note: Project must be created prior on the server before this command

Syntax

SetUserData(**pID** as integer, **vline** as integer, **name** as string) as integer

Parameter	**Description**
pID	Project identification number
vline	Agent extension
name	Agent name

SetWinTarget

Registers the client OCX with the window handle, so that when an unsolicited message arrived a message of **ClientMsg** will be posted to the window's queue specified by **hWnd**

Syntax

SetWinTarget(**hWnd** as Long, **ClientMsg** as Long)

Parameter	**Description**
HWnd	Window handle
ClientMsg	User define windows message

ShutDown

Terminates TCP connection and shutdown TCP stack

Syntax

ShutDown()

StartVL

Start recording the conversation to the voice file

Syntax

StartVL(**AgentID** as string, **Filename** as string, **MaxTime** as integer, **MaxSilence** as integer) as integer

Parameter	**Description**
AgentID	Agent name
Filename	Voice file name
MaxTime	Max time to record
MaxSilence	Maximum silence time

StopVL

Stops recording the conversation to the voice file

Syntax
StartVL(**AgentId** as string) as integer

Parameter	**Description**
AgentId	Agent name

TransferAgent
Transfers the call to another ready agent

Syntax
TransferAgent (**name** as string) as integer

Parameter	**Description**
name	Agent Name

XDial
Dials a number from a line in the specified project (group)

Syntax
XDial(**Proj** As Integer, **number** As String) As Integer

Parameter	**Description**
Proj	Project number
number	The number to dial out manually

Client Events

CallArrived
A call has arrived or will be assigned to this agent

Syntax
CallArrived(ByVal **ProjID** As Integer, ByVal **CallData** As String)

Parameter	**Description**
ProjID	Project number
CallData	Call data to be associated with the call

MsgFrCtrl
Message send from the control received

Syntax
MsgFrCtrl(ByVal **Msg** As String)

Parameter	Description
Msg	Content of the message

Monitoring Functions

ChgMonitorAttr

Supervisor is coaching or/not coaching the trainee on the phone

Syntax
ChgMonitorAttr(**AgentID** as string, **NewAttr** as integer) as integer

Parameter	**Description**
AgentID	Agent name
NewAttr	0 – Coach 1 – Not coach

MonitorAgent

Starts a monitoring section

Syntax
MonitorAgent(**AgentID** as string) as integer

Parameter	**Description**
AgentID	Agent name

UnMonitorAgent

Stops a monitoring section.

Syntax
UnMonitorAgent(**AgentID** as string) as integer

Parameter	**Description**
AgentID	Agent name